

---

# NSUserDefaultsController Class Reference

[Cocoa](#) > [Data Management](#)



2006-05-23



Apple Inc.  
© 2006 Apple Computer, Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY**

**DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## **NSUserDefaultsController Class Reference 5**

---

|   |    |
|---|----|
| Overview                                | 5  |
| Tasks                                   | 5  |
| Obtaining the Shared Instance           | 5  |
| Initializing a User Defaults Controller | 5  |
| Managing User Defaults Values           | 6  |
| Class Methods                           | 6  |
| sharedUserDefaultsController            | 6  |
| Instance Methods                        | 7  |
| appliesImmediately                      | 7  |
| defaults                                | 7  |
| hasUnappliedChanges                     | 7  |
| initialValues                           | 8  |
| initWithDefaults:initialValues:         | 8  |
| revert:                                 | 8  |
| revertToInitialValues:                  | 9  |
| save:                                   | 9  |
| setAppliesImmediately:                  | 10 |
| setInitialValues:                       | 10 |
| values                                  | 10 |

---

## **Document Revision History 13**

---

## **Index 15**

---



# NSUserDefaultsController Class Reference

---

|                            |  |
|----------------------------|--|
| <b>Inherits from</b>       | NSController : NSObject                        |
| <b>Conforms to</b>         | NSCoding (NSController)<br>NSObject (NSObject) |
| <b>Framework</b>           | /System/Library/Frameworks/AppKit.framework    |
| <b>Availability</b>        | Available in Mac OS X v10.3 and later.         |
| <b>Companion guide</b>     | Cocoa Bindings Programming Topics              |
| <b>Declared in</b>         | NSUserDefaultsController.h                     |
| <b>Related sample code</b> | Dicey<br>TemperatureTester                     |

## Overview

NSUserDefaultsController is a Cocoa bindings compatible controller class. Properties of the shared instance of this class can be bound to user interface elements to access and modify values stored in NSUserDefaults.

## Tasks

### Obtaining the Shared Instance

+ [sharedUserDefaultsController](#) (page 6)

Returns the shared instance of NSUserDefaultsController, creating it if necessary.

### Initializing a User Defaults Controller

- [initWithDefaults:initialValues:](#) (page 8)

Returns an initialized NSUserDefaultsController object using the NSUserDefaults instance specified in *defaults* and the initial default values contained in the *initialValues* dictionary.

## Managing User Defaults Values

- [defaults](#) (page 7)  
Returns the instance of NSUserDefaults in use by the receiver.
- [setInitialValues:](#) (page 10)  
Sets the receiver's initial values to *initialValues*.
- [hasUnappliedChanges](#) (page 7)  
Returns whether the receiver has user default values that have not been saved to NSUserDefaults.
- [initialValues](#) (page 8)  
Returns a dictionary containing the receiver's initial default values.
- [setAppliesImmediately:](#) (page 10)  
Sets whether any changes made to the receiver's user default properties are saved immediately.
- [appliesImmediately](#) (page 7)  
Returns whether any changes made to bound user default properties are saved immediately.
- [values](#) (page 10)  
Returns a key value coding compliant object that is used to access the user default properties.
- [revert:](#) (page 8)  
Causes the receiver to discard any unsaved changes to bound user default properties, restoring their previous values.
- [revertToInitialValues:](#) (page 9)  
Causes the receiver to discard all edits and replace the values of all the user default properties with any corresponding values in the *initialValues* (page 8) dictionary.
- [save:](#) (page 9)  
Saves the values of the receiver's user default properties.

## Class Methods

### sharedUserDefaultsController

Returns the shared instance of NSUserDefaultsController, creating it if necessary.

```
+ (id)sharedUserDefaultsController
```

#### Discussion

This instance has no initial values, and uses `[NSUserDefaults standardUserDefaults]` to create the defaults. An application can get this object when an application launches and configure it as required.

#### Availability

Available in Mac OS X v10.3 and later.

#### Related Sample Code

Dicey

#### Declared In

NSUserDefaultsController.h

## Instance Methods

### **appliesImmediately**

Returns whether any changes made to bound user default properties are saved immediately.

- (BOOL)appliesImmediately

#### **Discussion**

Default is YES.

This property is observable using key-value observing.

#### **Availability**

Available in Mac OS X v10.3 and later.

#### **See Also**

- [setAppliesImmediately:](#) (page 10)

#### **Declared In**

NSUserDefaultsController.h

### **defaults**

Returns the instance of NSUserDefaults in use by the receiver.

- (NSUserDefaults \*)defaults

#### **Discussion**

This property is observable using key-value observing.

#### **Availability**

Available in Mac OS X v10.3 and later.

#### **Declared In**

NSUserDefaultsController.h

### **hasUnappliedChanges**

Returns whether the receiver has user default values that have not been saved to NSUserDefaults.

- (BOOL)hasUnappliedChanges

#### **Discussion**

This property is observable using key-value observing.

#### **Availability**

Available in Mac OS X v10.4 and later.

#### **See Also**

- [appliesImmediately](#) (page 7)

- [setAppliesImmediately:](#) (page 10)

#### Declared In

NSUserDefaultsController.h

## initialValues

Returns a dictionary containing the receiver's initial default values.

- (NSDictionary \*)initialValues

#### Discussion

These values are used when is no value found for the bound property in [defaults](#) (page 7).

This property is observable using key-value observing.

#### Availability

Available in Mac OS X v10.3 and later.

#### See Also

- [setInitialValues:](#) (page 10)

- [revertToInitialValues:](#) (page 9)

#### Declared In

NSUserDefaultsController.h

## initWithDefaults:initialValues:

Returns an initialized NSUserDefaultsController object using the NSUserDefaults instance specified in *defaults* and the initial default values contained in the *initialValues* dictionary.

```
- (id)initWithDefaults:(NSUserDefaults *)defaults initialValues:(NSDictionary *)initialValues
```

#### Discussion

If *defaults* is nil, the receiver uses [NSUserDefaults standardUserDefaults].

This method is the designated initializer.

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

NSUserDefaultsController.h

## revert:

Causes the receiver to discard any unsaved changes to bound user default properties, restoring their previous values.

```
- (void)revert:(id)sender
```



**Discussion**

The receiver invokes `discardEditing` on any currently registered editors. The *sender* is typically the object that invoked this method.

If `appliesImmediately` (page 7) is YES, this method only causes any bound editors with uncommitted changes to discard their edits.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- `revertToInitialValues:` (page 9)

**Declared In**

NSUserDefaultsController.h

**revertToInitialValues:**

Causes the receiver to discard all edits and replace the values of all the user default properties with any corresponding values in the `initialValues` (page 8) dictionary.

- (void)revertToInitialValues:(id)sender

**Discussion**

This effectively sets the preferences that a user can change to their “out-of-the-box” values. This method has no effect if initial values were not specified. The *sender* is typically the object that invoked this method.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- `initialValues` (page 8)

- `revert:` (page 8)

**Declared In**

NSUserDefaultsController.h

**save:**

Saves the values of the receiver’s user default properties.

- (void)save:(id)sender

**Discussion**

This method has no effect if `appliesImmediately` (page 7) returns YES.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSUserDefaultsController.h

## setAppliesImmediately:

Sets whether any changes made to the receiver's user default properties are saved immediately.

- (void)setAppliesImmediately:(BOOL)flag

### Discussion

The default is YES.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [appliesImmediately](#) (page 7)

### Declared In

NSUserDefaultsController.h

## setInitialValues:

Sets the receiver's initial values to *initialValues*.

- (void)setInitialValues:(NSDictionary \*)initialValues

### Discussion

These values are used when a user default properties has no value in NSUserDefaults and by [revertToInitialValues:](#) (page 9).

The initial values must be set before loading a nib that uses the receiver, as those values may be referenced at load time. It is good practice to set the initial values—along with registering any defaults for the applications—in the `initialize` class method of your preference dialog controller, or the application delegate.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [defaults](#) (page 7)

- [initialValues](#) (page 8)

### Declared In

NSUserDefaultsController.h

## values

Returns a key value coding compliant object that is used to access the user default properties.

- (id)values

### Discussion

If present the value for the property in [defaults](#) (page 7) is returned, otherwise a corresponding value in [initialValues](#) (page 8) is returned.

This property is observable using key-value observing.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSUserDefaultsController.h



# Document Revision History

---

This table describes the changes to *NSUserDefaultsController Class Reference*.

| Date       | Notes   |
|------------|---|
| 2006-05-23 | First publication of this content as a separate document. |

**REVISION HISTORY**

Document Revision History

# Index

---

## A

---

appliesImmediately [instance method 7](#)

## D

---

defaults [instance method 7](#)

## H

---

hasUnappliedChanges [instance method 7](#)

## I

---

initialValues [instance method 8](#)  
initWithDefaults:initialValues: [instance method 8](#)

## R

---

revert: [instance method 8](#)  
revertToInitialValues: [instance method 9](#)

## S

---

save: [instance method 9](#)  
setAppliesImmediately: [instance method 10](#)  
setInitialValues: [instance method 10](#)  
sharedUserDefaultsController [class method 6](#)

## V

---

values [instance method 10](#)